

**AFRL-IF-RS-TR-2002-271**  
**Final Technical Report**  
**October 2002**



# **ASYNCHRONOUS TRANSFER MODE (ATM) SENTINEL INTRUSION DETECTION**


**General Dynamics**


*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**AIR FORCE RESEARCH LABORATORY  
INFORMATION DIRECTORATE  
ROME RESEARCH SITE  
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-271 has been reviewed and is approved for publication.

APPROVED:   
NELSON ROBINSON  
Project Engineer

FOR THE DIRECTOR:   
WARREN H. DEBANY, Technical Advisor  
Information Grid Division  
Information Directorate

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <b>OMB No. 074-0188</b>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> October 2002	<b>3. REPORT TYPE AND DATES COVERED</b> Final Feb 98 – Sep 01	
<b>4. TITLE AND SUBTITLE</b> ASYNCHRONOUS TRANSFER MODE (ATM) SENTINEL INTRUSION DETECTION			<b>5. FUNDING NUMBERS</b> C - F30602-98-C-0097 PE - 61102F PR - 2301 TA - 02 WU - 02	
<b>6. AUTHOR(S)</b> Robert N. Smith & Doug Hill				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> General Dynamics 8220 East Roosevelt Road Scottsdale Arizona 85252-9040			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Air Force Research Laboratory/IFGB 525 Brooks Road Rome New York 13441-4505			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>  AFRL-IF-RS-TR-2002-271	
<b>11. SUPPLEMENTARY NOTES</b>  AFRL Project Engineer: Nelson P. Robinson/IFGB/(315) 330-4110/ Nelson.Robinson@rl.af.mil				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				<b>12b. DISTRIBUTION CODE</b>
<b>13. ABSTRACT (Maximum 200 Words)</b> This effort studied civilian/military ATM networks concerning their vulnerability to several different classes of insider/outsider attacks, and then used this information to design and demonstrate a Sentinel that would mitigate all/some of those classes. The classes considered here were Denial of Service (DoS), Insertion of Covert Channels, traffic corruption, and Man-in-the-Middle, with a concentration here on DoS. A subcontractor, Prof Sumit Ghosh and several grad students of ASU, used ATMSIM 1.0 to develop a behavior model based on a 9 node/26 link ATM Network. ATMSIM 1.0 is a dynamic, asynchronous, distributed simulation that handles up to 30 nodes, and a license was obtained from DSP Inc. for 1 year to do this. The behavior model that was developed from this was used to gain insight about how the ATM Sentinel prototype would detect attacks, define signatures, and manage the network under stress from attacks. The ATM Sentinel prototype that was developed is not a firewall, but more like the behavior model used to define it, as this allows for less of a performance penalty (ie: allows higher throughput). The ATM Sentinel is located on the protection boundary of a network and uses parameters already measured by the ATM network to establish its metrics, thus saving valuable time. So, normal QoS conditions were defined and tolerance bands based on 1-sigma were established. These are displayed for a network operator or security analyst. Once the display shows the real-time QoS metrics leaving the tolerance band, an attack is detected and appropriate action can be taken. Previous simulation of the effect on the QoS, or tolerance bands, of the network of the different classes of attacks also allows for determining what class of attack is occurring.				
<b>14. SUBJECT TERMS</b> Asynchronous Transfer Mode-Sentinel, ATM-S, Quality of Service, QoS, Behavioral Model, Local Area Network Emulation, LANE, Network-Network Interface, NNI, User-Network Interface (UNI), ATM Simulation, ATMSIM 1.0				<b>15. NUMBER OF PAGES</b> 30
				<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b>  UNCLASSIFIED	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b>  UNCLASSIFIED	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b>  UNCLASSIFIED	<b>20. LIMITATION OF ABSTRACT</b>  UL	

## Table of Contents

1	Introduction.....	1
2	System Concept .....	2
2.1	Initial Studies .....	2
2.2	Operational Model .....	4
2.3	Implementation .....	6
3	Behavioral Model.....	7
3.1	Model Development.....	7
3.2	ATM attacks.....	8
3.3	Modeling Results .....	9
3.3.1	ASU Analysis.....	9
3.3.2	Conclusions.....	11
3.4	Issues.....	11
4	Prototype.....	12
4.1	Prototype Design.....	12
4.1.1	ATM Security Overview.....	13
4.1.2	Prototype Goals.....	15
4.2	Prototype Implementation.....	15
4.2.1	Objectives .....	15
4.2.2	Prototype Processing.....	17
4.3	Prototype Results .....	18
5	Summary.....	21
5.1	Findings.....	21
5.2	Future Work.....	21
5.2.1	Follow-on activities .....	21
5.2.2	ATM-Sentinel Development.....	22
	Appendix A Prototype Display Development .....	23

## List of Figures

Figure 1. The ATM Sentinel concept includes monitoring of the message passing the boundary of the enclave as well as monitoring the performance of the network. ....	3
Figure 2. The near term ATM-Sentinel provides monitoring of an enclave of hosts and links, such as a building or base. ....	4
Figure 3. The system level concept includes hierarchical protection. ....	5
Figure 4. The performance analyzer is part of operator’s visualization station.....	6
Figure 5 The simulated network includes three peer groups with a total of 9 nodes and 26 links. ....	7
Figure 6. The statistics for attack 3 provide a recognizable signature for an intrusion detector to identify the presence of the attack. ....	12
Figure 7 ATM-S nested security association for secure communications via NNI and UNI.....	13
Figure 8 Top-level context diagram of the ATM sentinel system of central and distributed components .....	14
Figure 9 Top Level Prototype Diagram .....	16
Figure 10 Explanation of ASU Network Data Files .....	17
Figure 11. Sample Data: Bandwidth as a Function of Time.....	18
Figure 12. Prototype processing steps .....	19
Figure 13 Node Balance Calculation Includes Data In, Out, Source, and Sink at a Node .....	23
Figure 14 Display of Node2 Balance: Training (Normal) Data .....	24
Figure 15 Training (Normal) Data with tolerance bands.....	24
Figure 16 Real-Time Data with Threshold Crossing.....	25

## List of Tables

Table 1. Link data for simulation nodes .....	8
---	---

# **ATM-Sentinel Final Report**

## **1 Introduction**

The ATM Sentinel project comprised three phases. The first phase was a review of relevant ATM protocol and security documents to determine the state of the art and develop a concept for the system. During this phase we also identified attack scenarios that were specific to components of the ATM PNNI specification and that could result in serious degradation of an ATM network. These attack scenarios were provided to our subcontractor, Professor Sumit Ghosh of Arizona State University, to develop specific attack details for analysis in their behavioral model. That model allowed us to gain the insight necessary to detect the attacks and define signatures for the ATM-Sentinel prototype.

The second phase entailed the development of the behavioral model and the generation of data from it using a simulation developed by Prof. Ghosh and his students to study dynamical, asynchronous systems. In this phase of the program, Prof. Ghosh studied variations on five different attacks. He considered different load levels placed on the network by the attacker. He also studied the effects of target or link location in the network on the effective of the attack on the target and on the network as a whole. His results showed that location does affect the network statistics. He also showed that the signature of the attack may be distributed around the network and it is apparent not only in performance degradations, but also performance improvements. That is, some of the nodes or links may actually have better performance because the attack reduces the load that reaches them.

The third phase of the program took the results from the ASU modeling effort as the basis for defining a prototype implementation of the ATM-Sentinel. That prototype included code to parse and analyze the data files from ASU into a form that could be displayed for a network operator or security analyst. Using the parsed data, we first defined normal traffic and tolerance bands about the normal based on 1-sigma variations of the observed data. Once the tolerance bands were defined, we plotted them, and the simulated operational, real-time data that the operator would see monitoring the network. We defined an attack as the real time data leaving the tolerance bands and demonstrated that the attacks could be detected given the proper measurements for the attacks and definition of tolerance bands.

## **2 System Concept**

When the program started, we had to resolve a difference of opinion as to the primary approach for ATM-Sentinel. On the one hand, several believed that the program should be primarily a firewall for an ATM enclave. Others believed that because of the small cell size and specialized addresses in the header a simple firewall would not offer sufficient protection. The behavioral model that would be developed by ASU was the proper tool to resolve the issue. The Statement of Work stated that we should investigate attacks in the order of Denial of Service, Insertion of Covert Channels, Traffic Corruption and Private Network Search and Modeling. Due to the limited resources, we agreed to restrict our efforts to the Denial of Service (DoS) issue.

We also noted that many of the parameters that firewalls traditionally use to filter incoming messages were masked inside the payloads of several ATM cells. Therefore, a firewall concept was inadequate unless it could reassemble messages for each of the channels entering the enclave and would lead to too large a performance penalty for high speed networks. Thus, we established the need for a broader solution. That is, we would concentrate on the parameters already measured by ATM networks in establishing their Quality of Service (QoS) metrics and look for the effects of attacks on those metrics. We then investigated various ATM protocols and identified some attacks that would be specific to ATM rather than IP and would, if successful, lead to reduced service for the network.

We also developed an initial implementation concept to guide the studies and behavioral model development as they progressed. That concept monitored the boundary of the enclave and processed QoS data from the various switches interior to the enclave. The monitor was implemented with a processor that could, if necessary, also decrypt the cells or cell payloads to look for suspicious addresses during routing and briefly monitor the payloads for specific items such as bad words. The intent was that the monitor would not have to reassemble the total message, but rather act as a first line of defense in a layered system.

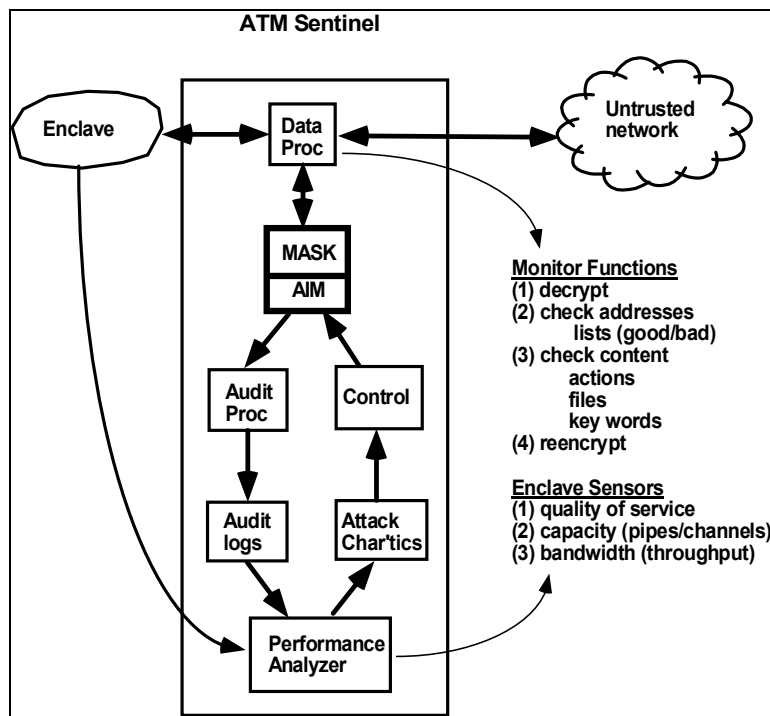
### **2.1 Initial Studies**

At the onset of the project, the ATM Forum documents were reviewed for applicability to the ATM Sentinel project. The review focused on the Anchorage accord document set that is generally considered the baseline, with additional ATM Forum and ITU documents referenced by the Anchorage accord documents. These reviews focused on establishing the requirements for the behavioral model simulation and for the definition of attacks specific to the ATM protocols.

A definition of the target environment has been evolving. The focus of attack studies has centered on the PNNI (Version 3.1) protocols at this time. Since the end-to-end routing and signaling is done according to the PNNI specification and since PNNI is becoming the recognized standard for private to private and private to public control-plane, the PNNI protocols are the first place to start. Attacks against other ATM structures such as LANE, MPOA, and the ATM header were deferred for future evaluation. LANE and MPOA were deferred because they represent specific implementations supported by ATM and our focus was on more general ATM attack issues. Attacks to the 5-byte ATM cell headers were deferred for another period because of the similar MAC & Physical layer attacks being covered by work done on other lower layer protocols. Attacks on the headers would be detectable by Quality-of-Service (QoS) monitors at either end of the path.

We defined specific attacks based on the ATM protocols. Scenarios that include protocol-specific denial-of-service, masquerading, and eavesdropping were considered with priority given to DoS. Prioritization of attack scenarios was determined based on the likelihood that an attacker could obtain insider access to initiate the attack. Outsider attacks were considered but with reduced priority because ATM cell-body encryption will provide a degree of protection against them. Attacks to the ATM payload data destined for the layers above the datalink layer would be detected or blocked by current network firewall technology. If a denial-of-service were waged by randomly changing bits of the ATM cell, the ATM metrics would sense a drop in quality-of-service. An attack waged against ATM routing or call-setup transmissions could be difficult to detect; hence, that was an important focus of this effort.

The behavioral model considered attack scenarios waged against the PNNI routing and signaling packets and messages. Particular PNNI packet parameters were identified that, when modified by an attacker, would lead to denial-of-service, could allow a third party to eavesdrop, or would allow the attacker to masquerade as an authorized node. The behavioral model was implemented on an ATM PNNI testbed and the results were used in the ATM-Sentinel prototype.



**Figure 1. The ATM Sentinel concept includes monitoring of the message passing the boundary of the enclave as well as monitoring the performance of the network.**

Several simulators were reviewed for applicability to the ATM Sentinel project. MIL3's OPNET and the NIST ATM simulator were considered along with the one at Arizona State

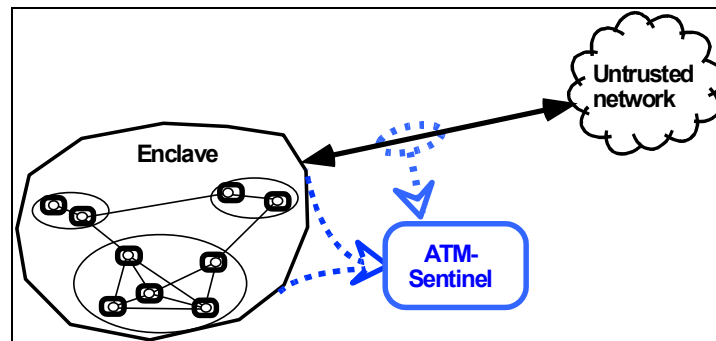


University. The ASU simulator was selected because of the proximity of the simulation developers and of the simulation's capability for parallel implementation.

## 2.2 Operational Model

In parallel with studying network attack methods and their effects, we considered the implementation of the ATM Sentinel. Our current thinking shows an enclave protection system that provides both firewall and sniffing capability as shown in Figure 1. Data entering or leaving the enclave are interrogated as to their source or destination as maintained in the routing information that was used to set up the connection. If the addresses change, the monitor will alert the enclave manager and implement appropriate action. The ATM-S will also analyze the cell payloads to evaluate their contents and the possibility of proscribed information passing in either direction. This may or will require that certain elements of the ATM SAR process be implemented in the firewall as well as at interior nodes. It may also require encryption/decryption of the cells as they pass through. Additional processing of the enclave performance statistics, such as quality of service, available capacity, and remaining bandwidth on the various links in the enclave would be monitored for unusual changes. Whenever an ATM Sentinel raises an alarm, it will store the details of the information that created the alarm and any subsequent actions.

The operational model that we propose for the ATM-Sentinel implementation evolved as the project continued and the ASU results became available. ASU's work indicated that detection of ATM attacks through monitoring of network Quality of Service (QoS) parameters will require monitoring of both the external connection from the protected network and of all nodes and links in the protected network. Monitoring of the external connection allows identification of specific addresses and connections or signaling based attacks. Monitoring of the QoS on internal nodes and links allows identification of attacks from either insiders or outsiders who have subverted internal nodes. The simulation results indicate that all monitoring of all the nodes and links is necessary because some attacks will only become apparent in their effects on remote nodes. Furthermore, the attacks may be revealed by apparently better performance at remote nodes due to reduced traffic from the attacked node.

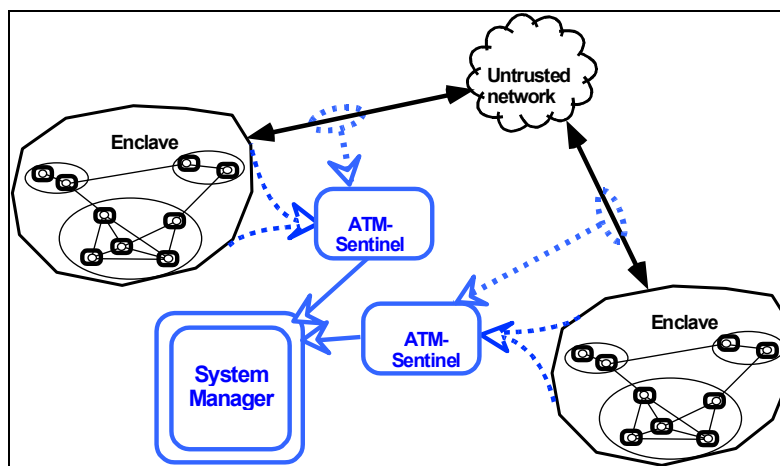


**Figure 2. The near term ATM-Sentinel provides monitoring of an enclave of hosts and links, such as a building or base.**

Figure 2 shows a near term implementation vision for ATM-S protecting an enclave. The dashed lines indicate the monitoring of both the external connection and the internal nodes and links. Those data are transmitted to the ATM-S manager for detecting attacks and reacting when appropriate. The manager should be external to the protected network so that it is protected from the normal traffic on the network. Ideally, the manager would communicate via some separate channel so that an intruder would not be able to use the monitoring and management signals to map the network or infer when he had been detected.

The long-term vision in Figure 3 shows ATM-S protecting a network of enclaves hierarchically. In this case, a local ATM-S monitors and manages each enclave. The enclave managers then communicate the significant events and actions to an infrastructure or system manager. The system manager detects distributed attacks, reacts by warning the lower level enclaves when specific attacks are seen on portions of the network, and manages system-wide protection parameters.

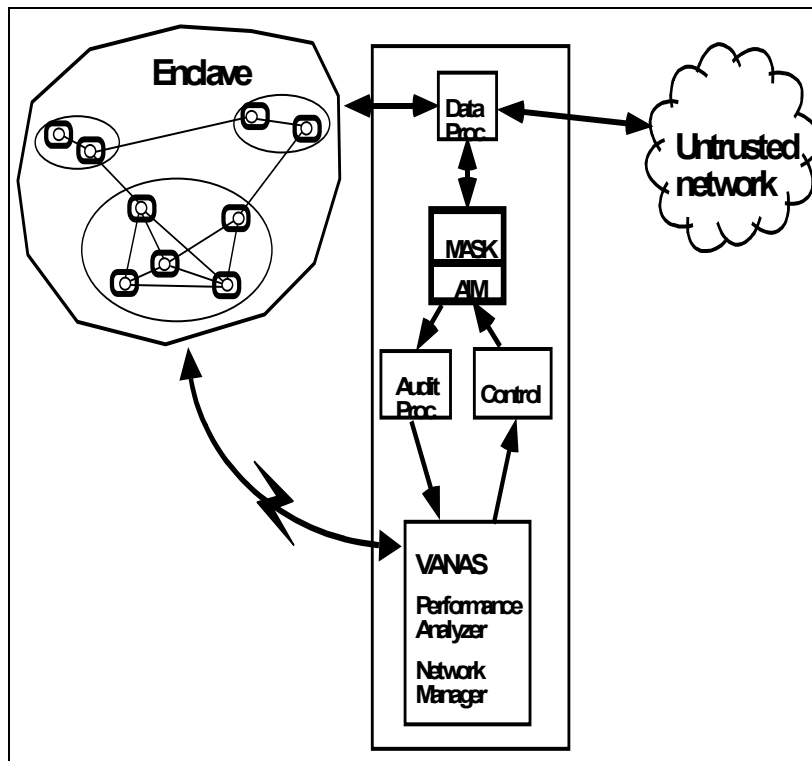
As the results from the ASU contract became available, we modified the design concept for ATM-Sentinel. Figure 4 shows the revised concept with a wireless side-channel for security data, using the ASU network as an example. The most significant finding of the ASU work is that the monitoring must cover the entire network, not just target nodes, and certainly not just the boundaries between the protected enclave and the outside world. Thus, we show each node of the network with a protection and monitoring element, all of which communicate with the network management and control system. For that system, we show a VANAS (Visualization and Analysis of Network Attack Status) management system communicating with the network protection elements via a side channel. The VANAS concept was invented and initially developed by Motorola. It continued some development with an agency under a government contract. That contract has ended and it is continuing development with Motorola funding as the Motorola Intrusion Vision. It provides monitoring and display of computer time views of the network events to detect attacks, react to them, and increase the network protection as the need arises.



**Figure 3. The system level concept includes hierarchical protection.**

## 2.3 Implementation

The results of the behavioral model confirmed that, for the attacks selected, monitoring of the QoS variables could indeed identify changes in network performance due to various types of DoS attacks. Those variables included:



**Figure 4. The performance analyzer is part of operator's visualization station.**

Available bandwidth,

Fraction of calls completed,

Average set-up time - outgoing calls,

Average latency - incoming calls, and

Cell loss ratio.

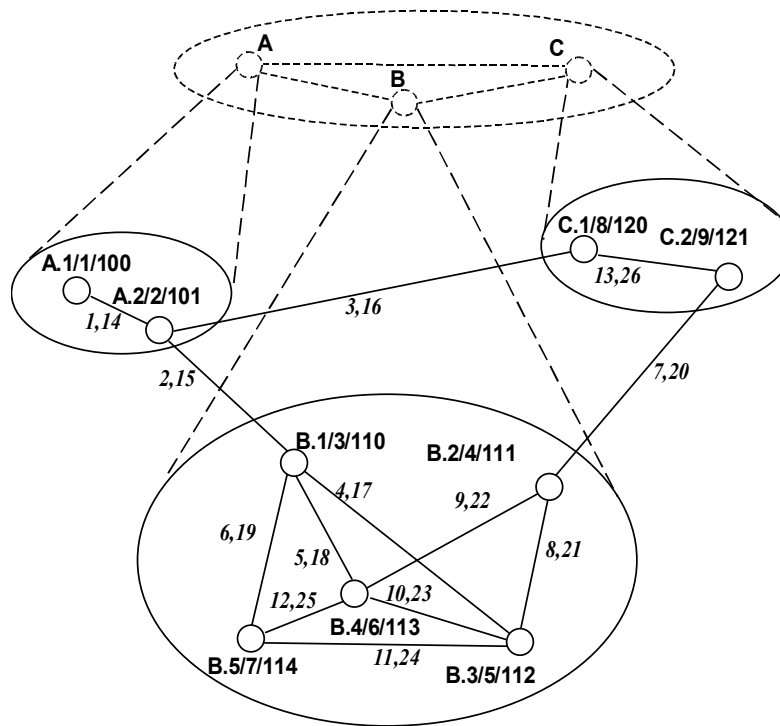
The basic ATM-Sentinel would monitor the network statistics from all of the switches in the enclave and compare them to the expected or normal traffic flow statistics to identify when conditions changed. The monitor would display these data and bands of acceptable variation to identify potential attacks. The network manager would then verify the event as either an attack or an unusual, but acceptable, level of performance due to some other cause.

### 3 Behavioral Model

As part of the effort to understand ATM attacks, we subcontracted with Professor Sumit Ghosh of Arizona State University to expand an existing ATM simulation to handle several attacks that we would mutually define. The model contained the call processor, the signaling network, the switch fabric, and the traffic network. The simulation is time based with the time step size set by the ATM cell transport through the switch fabric, which is the fastest sub-process in the system. Where possible the implementation followed the PNNI specification. Some of the necessary specifications were not completely developed in the PNNI documentation; for those, Prof. Ghosh developed extensions to enable the simulation to be completed.

Once the model was completed, Prof. Ghosh took some attack scenarios provided by Motorola and defined the details of attacks to be studied with the behavioral model, hosted on his simulation. Those attacks were implemented in the behavioral model and then into the simulator. Once each attack was implemented, it was analyzed with the simulator in a series of runs that started with normal traffic load on the network until it reached equilibrium. After the network reached equilibrium, the attack was launched and the network statistics were collected and compared to similar statistics from totally normal runs to identify the presence of the attack.

#### 3.1 Model Development



**Figure 5** The simulated network includes three peer groups with a total of 9 nodes and 26 links.

The license for the basic simulation and the number of computers that were available limited the size of the network for the study. Professor Ghosh developed a small network (see Figure 5) comprising 9 nodes in 3 groups and collected results from five scenarios in which the attacker attempts to block communication from one of the nodes to a different node. The figure identifies the nodes in the form A.x/y/zzz where the A indicates the peer group, the number x indicates the component of that peer group and the number y indicates the global node identifier. The triplet zzz indicates the logic identifier for the node.

The network consists of the 9 nodes and 26 (one-way) links between them. To provide realism, the physical locations of the nodes were taken from previous work by Prof. Ghosh's group. From the physical location, the simulation then established a time delay based on fiber optic connections and converted those delays into time steps in the simulation. Thus, when a message is transmitted from one node to another, it includes the inherent delay for transmission time. Table 1 provides the details of these links.

**Table 1. Link data for simulation nodes**

Link, $x \leftrightarrow y$	Link ID, $x \Rightarrow y$	Link ID, $y \Rightarrow x$	Distance	Delay (time)	Delay (steps)
A1 – A2	1	14	1158.16	5.79	2121
A2 – B1	2	15	2369.90	11.84	4337
A2 – C1	3	16	4420.08	22.1	8095
B1 – B3	4	17	2405.78	12.03	4407
B1 – B4	5	18	927.91	4.64	1700
B1 – B5	6	19	1486.07	7.43	2722
B2 – C2	7	20	1546.73	7.73	2832
B2 – B3	8	21	1268.37	6.34	2322
B2 – B4	9	22	356.23	1.78	652
B3 – B4	10	23	1505.06	7.5	2747
B3 – B5	11	24	1916.00	9.58	3509
B4 – B5	12	25	1347.38	6.74	2469
C1 – C2	13	26	1022.04	5.11	1872

For this effort, Prof. Ghosh's procedure tested various network load levels up to the point where the network goes unstable in performance. He then drops the load somewhat and runs his experiments at that level. This allows him to see the effect of changes on network performance due to the attack more quickly than if the load was at a more typical low utilization level for the normal traffic load. The simulation time divides into three phases: network establishment, growth to steady state, and the experiment period. The network establishment period could take a significant percentage of the total period he was using for these tests (20-40%). The time for load stabilization is "very quick." The third period is the time when the attack is presented to the network and the change in statistics is observed.

### **3.2 ATM attacks**

Professor Ghosh, worked with 5 attacks that, if successful, would result in an effective denial of service to the attacked node, or to the network in general. Two involve violation of the

assumptions of correct performance for a node in an ATM network. The other three represent excess consumption of resources, either in large chunks or small, and either in general or targeted at a specific node or network link.

- Attack 1: The attacker accepts call setup requests from the target node and then drops the connection without advising the target node. Thus, the node will continue sending cells as though the connection was successful, but the cells will be trapped by the attacker.
- Attack 2: After a connection is established with a target node, all cells directed either to or from that node will be redirected to an arbitrary and randomly changed path.
- Attack 3: The attacker will create dummy sessions to consume network resources, either requesting connections with large bandwidth requirements or many connections with smaller requirements.
- Attack 4: Similar to attack 3, but with a specific node as the focus of the attack. Thus, the attacker will request channels to a specific node in an attempt to consume all of that node's resources.
- Attack 5: Similar to attack 3, but targeted at a specific link between two nodes.

### **3.3 Modeling Results**

#### **3.3.1 ASU Analysis**

For each attack type, approximately 30 runs of the simulation provided both checkout of the attack implementation and experimentation to understand the effects of the attack. This set then included the "production" runs whose results were formally reported in the August 18, 1999 oral review and the November 23, 1999 ASU Final Report, both of which have been previously documented. Each of these runs took approximately 10 hours on a simulation testbed consisting of three laptop PCs running the Linux operating system.

##### **Attack 1:**

Particular nodes were selected for the attacker and target. All cells after call set-up was completed from the target passing through the attacker node were deleted. The target node was an interior node of peer group B. The attacker was selected as the gateway between peer groups B and C. Thus, all calls from the target node to nodes in the C group would first attempt to go through the attacker node. The effect of the attack becomes manifest in the reduced percentage of successful calls from the target node. Because of bandwidth no longer used by the dropped cells, the number attempted may actually increase. Similarly, link utilization shows a varied pattern. Those links carrying calls from the target after the attack showed reduced utilization for the targeted calls compared to normal; therefore, there is increased availability for other calls. Consequently, some calls from other nodes that would have been blocked by the targeted calls can now complete using links carrying calls from the targeted node. We conclude that, if the attacker is careful to mask his activity from the target, the actual signature of the attack will be

distributed over the network. Further, the signature will be complex in that some statistics will increase and others will decrease.

#### Attack 2:

In this attack, calls from the target node are randomly redirected to other nodes after the call has been set up. The key measure is the number of calls dropped on various links. Again the effect is distributed and can be either increased or decreased depending on the relationship of the specific link to those carrying the traffic from the attacked node. Several call volumes from the attacked node were evaluated, but they did not affect the qualitative results for this attack type. Note that depending on the attacker's randomization pattern, the traffic statistics may either be more balanced or show periods of higher or lower mean usage than the long term normals.

#### Attack 3:

This attack represents a very general denial of service attack wherein the attacker generates session requests randomly in the network. Two attack variations were investigated. The first had the attacker requesting very large sessions that would consume a large fraction of a link if accepted, but had the chance of not being accepted due to the volume requested. The second attack type used more small volume requests to accomplish the same end. To maintain comparability the total volume in the requests was the same under either variation. Several different total load levels were considered to determine the effect of volume on the results. The primary figure of merit for this attack type is the fraction of successful calls. The results for this attack show the strongest effect as the attack grows from no volume to the specified volume. Beyond that, increasing volume does not have as large an impact, indicating that the network quickly becomes saturated. Typically it appears that the strategy of many small calls is less effective at reducing network performance at lower attack levels than a few large calls. At higher attack volumes, the many small calls, because they can more easily consume the remaining bandwidth are more effective.

#### Attack 4:

Attack type 4 is similar to Attack type 3, except that it focuses on a specific node instead of the network in total. Cases using different target nodes were investigated to determine the effect of the location of the node in the network on the attack success. The study considered whether the target was interior to the network or was a gateway to a different network. When the node is interior to the network, the effect of the attack is seen at both the target node and the attacker node. Thus, the attacker in effect also suffers a denial of service because he consumes his own resources in pressing the attack. Other nodes are also affected to the extent that they try to route messages through either the target or the attacker. When the target is a gateway node to a different network, the effect is reduced because the attacker has less access to the other side of the target's links.

#### Attack 5:

This attack is also similar to attacks 3 and 4, except that a specific connection link is the target rather than a specific node or the network in general. Again, the attacks are grouped into several cases that represent attacks on internal links or on links to external networks. The first case involves a link connecting two interior nodes with relatively low connectivity to other parts of the network. For this case, the attacker sends messages to both ends of the link to attempt to

completely shut down connectivity. The second case involves a link that connects an interior node to a gateway node and has much greater connectivity to the rest of the network. The third case involves a link that connects from a gateway node to an exterior network. Results for the first case show the most impact on the target link (both directions) and on those links that the attacker uses to reach the endpoints of the target link. The results for the second set are similar, with the impact on non-target links and nodes much greater due to the greater connectivity of the target node. The results for the third case show a reduced impact to the target link because the attacker has greater difficulty focusing an attack at the end that is in the other network.

### **3.3.2 Conclusions**

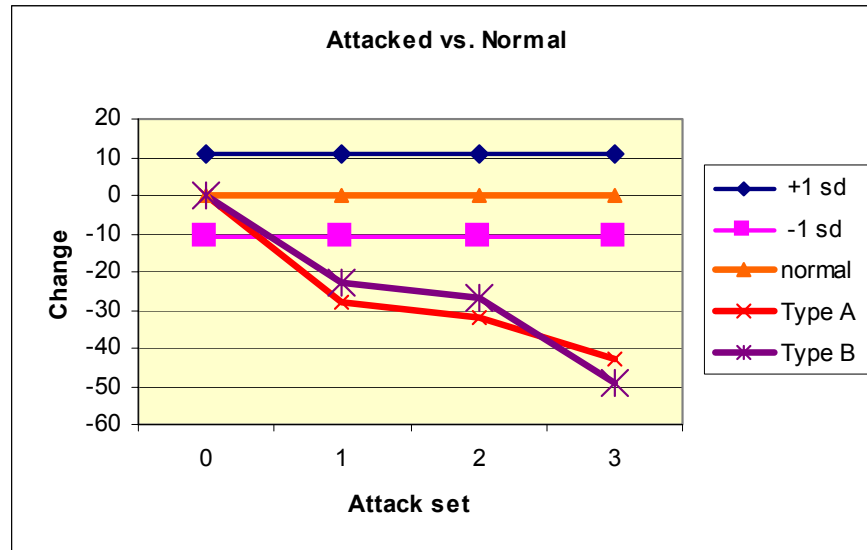
All of the test cases for the simulations showed that the characteristics of the attacks are distributed over the network, even when the attack is focussed on a particular node or link. Thus, the ATM-Sentinel implementation must involve sensors at each of the nodes to monitor Quality of Service metrics and communicate those metrics to a central station that can provide an analysis and comparison between the current data and the normal operation of the network. These conclusions seem consistent across the different conditions and conform to a reasonable model of network performance. However, the results are from a very small network and are based on a very few data points with very little sensitivity analysis of the quantitative values. Thus, they point the way, but need to be verified in a more detailed and rigorous modeling with a more detailed and complete capturing of the ATM protocols.

### **3.4 Issues**

The behavioral model must be continued and improved in several areas before its results can be used to set design parameters. A more complete implementation of the PNNI protocol, especially the routing and segmentation / re-assembly processes will allow a more complete understanding as to the total impact of network errors due to attacks. When the simulation is complete, a broader and more varied set of attacks needs to be developed to match whatever experience network operators have about susceptibilities and attacks. In addition to the improved modeling fidelity, more cases, both for normal conditions and under attack need to be generated so that a firm statistical base can be developed for the signatures that are selected for the ATM-S. The simulation should also run several different network configurations, sizes, and loads to determine where and how network design impacts target signatures and network protection. Additional data must also be developed to determine how ATM network loads and statistics vary with time of day and time of year factors.



## 4 Prototype



**Figure 6. The statistics for attack 3 provide a recognizable signature for an intrusion detector to identify the presence of the attack.**

To complete the project, we constructed a software prototype that would implement a system to illustrate the lessons learned from the behavioral model. We concentrated on attack 3 as it represents a broadly based attack that, if successful, would be very damaging to the network's ability to continue operation. Analyzing the data from ASU, we developed the chart in Figure 6 to indicate the signature of the attack. That chart shows five data lines. The nominal line, with the triangle markers, represents the typical load at each node of the network. The other two horizontal lines represent the tolerance bands for the variation of the nominal among nodes and over time. Thus, load measurements that occur within the region bounded by these lines would be considered typical network variation. The final two lines represent the network behavior under the two attack variations described in Section 3.3.1. Both attack types add the same total load to the network. The first has a few very large capacity requests, the second has many low capacity requests. As the plots show, the effect of the different ways of requesting additional data is small relative to the normal variability of the network data. However, the change as the attack level progresses shows a significant change that is outside of the normal variation and, thus, detectable.

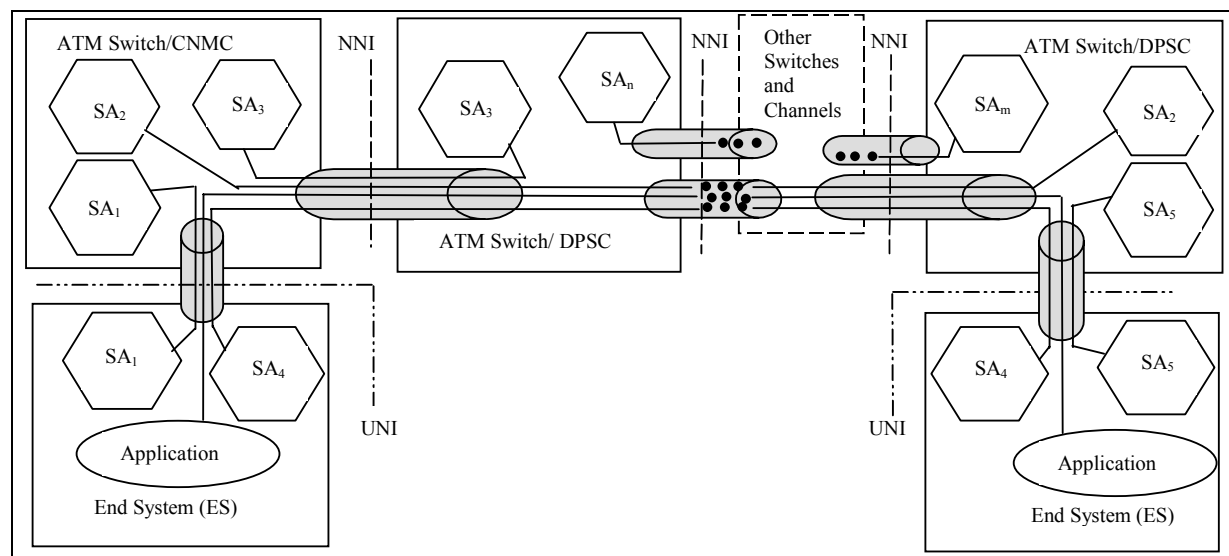
### 4.1 Prototype Design

The ATM-Sentinel concept architecture combines a central monitoring component and distributed switch agents. The central monitor will collect information from switches to determine the faults due to hardware failure or security attack. The ATM-S will provide network

security by monitoring/reporting of network parameters that will lead to the early detection of faults and attacks.

#### 4.1.1 ATM Security Overview

In the ATM Forum's view of security for ATM<sup>1</sup> networks the Private Network-to-Network Interface (PNNI) provides routing and switching for Network-to-Network Interface (NNI) as well as User to Network Interface (UNI). The UNI interface is normally at an End-System (ES), which can be a workstation or LAN. PNNI routing is primarily concerned with routing from one ES to another ES via an ATM network composed of many switches. The outside network may be composed of many ATM switches that operate by communicating across NNIs. If the ATM network is integrated with a public network, the public network is assumed to be untrusted and typically provides only permanent virtual circuits. Figure 7 shows ESs connected to the edge of an ATM network of switches. The switches connected to the End-Systems may act as Firewalls, while one of the ATM switches may act as the ATM-S management node. The ATM-S management node will be known as the Central Network Management Component (CNMC). Other ATM-S switches in the network will be called Distributed PNNI Switch Components (DPSC). Also shown in Figure 7 are Security Associations (SAs) between various nodes. Security associations are created to protect various security services provided by the nodes, such as authentication, data confidentiality, data integrity, or access control purposes. The SAs are shown as pairs to reflect two security agents protecting a single virtual channel or virtual path while communicating securely between one another.



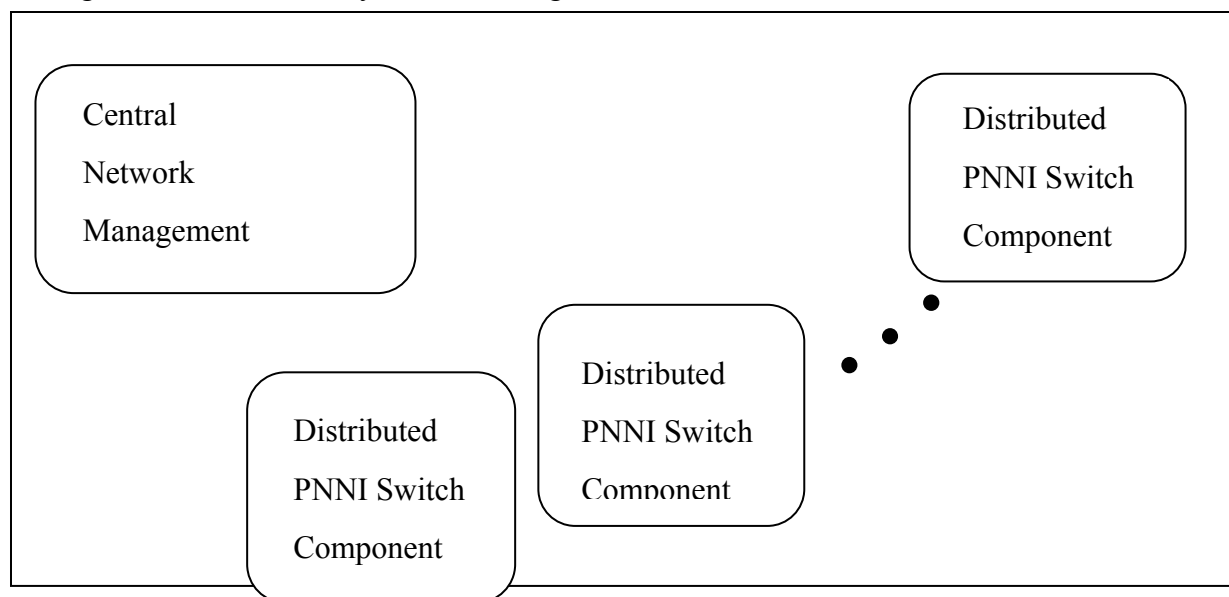
**Figure 7 ATM-S nested security association for secure communications via NNI and UNI**

<sup>1</sup> ATM Security Specification Version 1.0, February, 1999

The central network management component (CNMC) will provide the glue for end-to-end network security. The CNMC can interface with other CNMCs to allow for a hierarchical or distributed system at a higher level. The distributed PNNI switch component (DPSC) is the agent installed within the ATM network switches to collect and forward the necessary statistics.

The communication between trusted nodes of the system will be protected by the ATM Security Message Exchange protocol (SME). SME communication protocol will use security services implemented to support the security architecture defined by the ATM Forum Security Specification. The CNMC will communicate with trusted DPSCs to collect information relevant to health and security of the entire network. The central management node will provide coordination of distributed features (see note in Figure 8). Authentication, access control, and intrusion detection are integral features of the ATM-S. Multiple security services will be used to provide data integrity and confidentiality of management communication and user communication.

The ATM-S system components will communicate out-of-band using a three-way secure message exchange protocol. The integrity of each node will be protected by security support services that are integrated into each device. The integrated security device will provide basic services to include the three-way message exchange in the ATM protocol control plane. Encryption services (to include asymmetric and symmetric cryptography) will be provided for key exchange. The key management includes session key update and management of certificates. The functions of the security device will enable authentication and data integrity of the ATM-S management and user communication. Each peer level communication will be protected by separate security services. As noted in Figure 8, the ATM-S system is expected to be hierarchical. The central ATM-S management nodes will be able to report to higher-level management nodes that may be monitoring other LANs.



Note: A hierarchy of CNMCs will provide for a distributed security solution.

**Figure 8 Top-level context diagram of the ATM sentinel system of central and distributed components**

#### **4.1.2 *Prototype Goals***

The prototype phase had two major tasks. The first major task was the development of algorithm(s) to detect the attack(s) and display that information to an operator. The second major task was to develop a message simulator that represents 9 ATM switches communicating in a standard peer group configuration. One of the nodes acted as the monitor of the network. This simulator enabled the evaluation of the concept of a central processing node used to collect the data from the individual switches. The central monitor evaluated the data for the purpose of providing intrusion detection. Attack 3 as described in paragraph 3.2 and 3.3 above was the primary focus of the work during this phase.

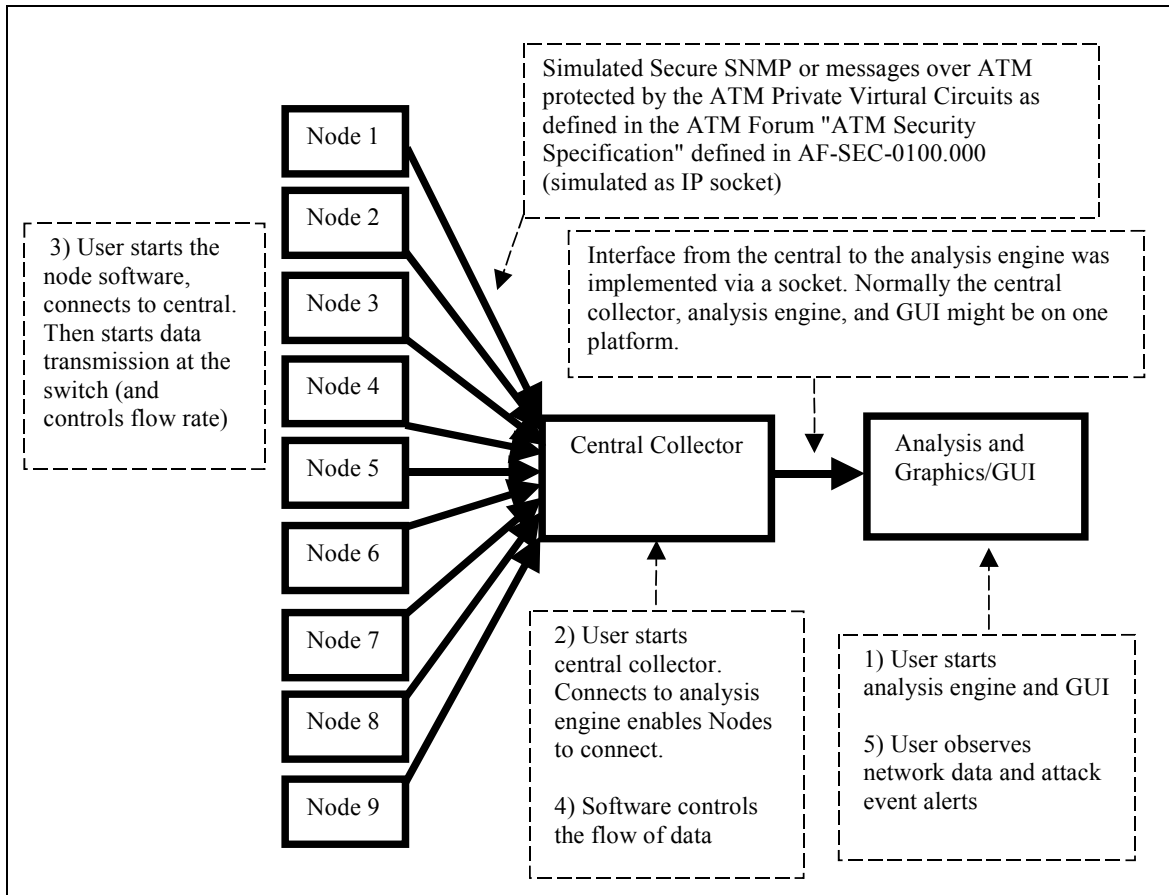
### **4.2 *Prototype Implementation***

#### **4.2.1 *Objectives***

- 1) Present the ATM-Sentinel architecture (in a prototype fashion) to protect ATM network.
- 2) Base the ATM-Sentinel prototype concept on the ASU behavioral model data.
- 3) Demonstrate how network data would alert an ATM network operator to attacks.

##### **4.2.1.1 *Prototype Architecture***

The ATM-Sentinel prototype models the architecture shown in Figure 7 and Figure 8. A common node processor was developed to simulate the ATM switch. The common node processor was used for switches identified as nodes 1 through 9 in Figure 9. The simulated network includes three peer groups with a total of 9 nodes and 26 links. The same software used for the simulated switch was also used for the central collection node. The central collection node receives data from each of the simulated switch nodes that read node balance data from a file associated with that node. The central collection node forwards the all of the node data it receives to the analysis engine. Details of the analysis engine follow in paragraph. Figure 9 shows this simplified block diagram of the software prototype.

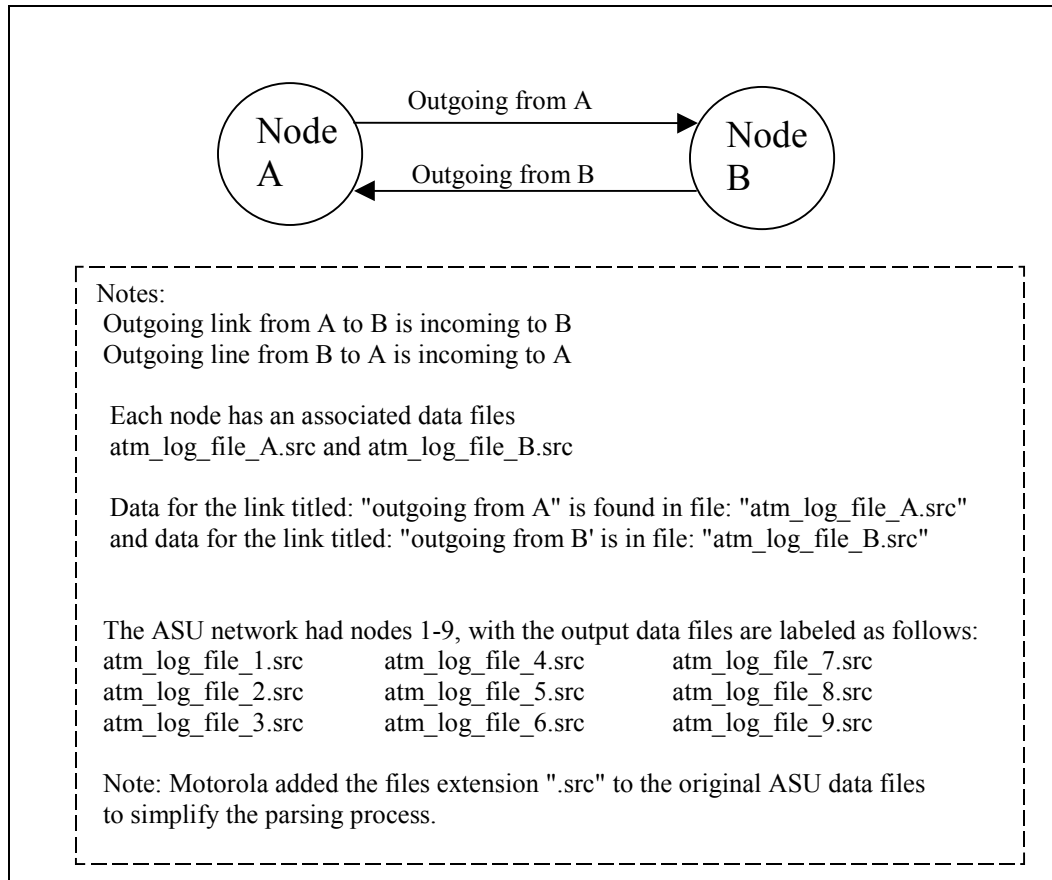


**Figure 9 Top Level Prototype Diagram**

#### *4.2.1.2 Exploitation of ASU behavioral model data*

The ASU behavioral model data represented 5 attacks as described in the November 1999 report by Dr. Sumit Ghosh. The simulated network included 9 ATM switches. Attack number 1 had 4 variations. Normal and attack data were provided. The normal data for attack number 1 were used as the normal data for all other attack scenarios. The prototype was tested with data from attack 3 of the ASU behavioral model study.

The attack data were saved by the ASU simulator into files associated with each of the 9 network nodes. Link bandwidth data for each node were stored in files according to each outgoing link. Thus, data flowing from node A to node B will be in a file labeled with "A," and data flowing the opposite direction will in a file labeled with "B," see Figure 10.



**Figure 10 Explanation of ASU Network Data Files**

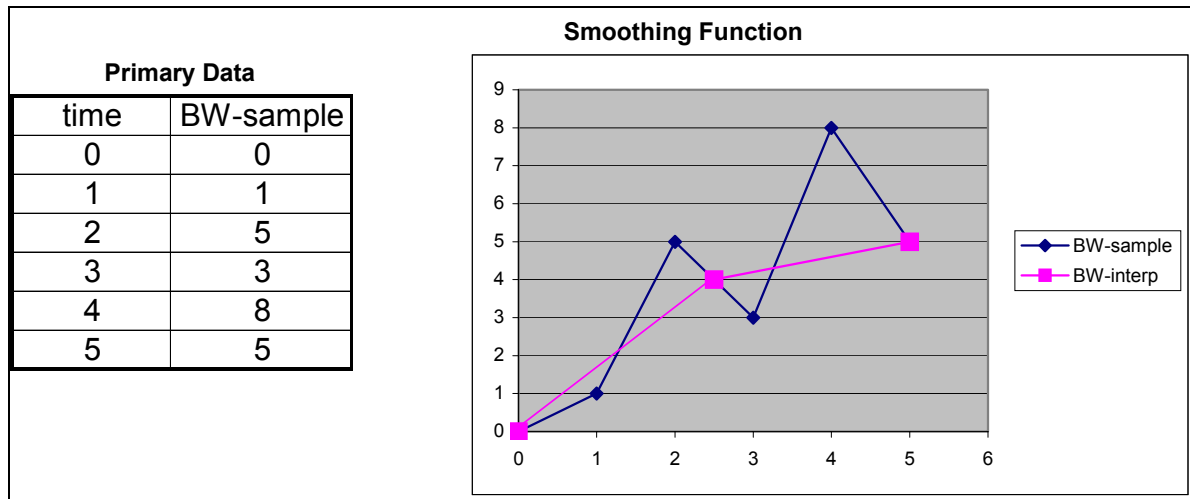
#### 4.2.2 Prototype Processing

The ATM-Sentinel prototype has two modes (or phases). One mode uses training data to generate the tolerance bands for the operational data. The second mode accepts data that simulate real-time operation containing attacks. A label on the input data identifies the mode of the input data.

The ASU Attack 1 "normal" data provide the basis for all of the tolerance bands for all of the nodes and attacks. The training consists of capturing many replicates of the normal data and computing the tolerance bands from the replicate statistics. In actual operation, the training would occur over many sets to statistically smooth the data and should include enough time to capture the normal seasonal and diurnal variation.

After the training data and tolerance bands are computed and stored, the real-time, operational data are compared to the tolerance bands. If the plot of the real-time data passes outside a tolerance band, an alert is issued. Because sample times in the operational data are different from those in the training set, the ATM-Sentinel demonstration uses a simple interpolation algorithm to fill in sample time points that do not exist in the training data set. As an example, we use simplified bandwidth data in figure 11 to demonstrate the algorithm. For this example, assume that the training data exist at integer times while the operational call times happen to be at 0, 2.5,

and 5. When the simulation time steps to  $t=2.5$ , the algorithm interpolates between the tolerance band data values at  $t=2$  and  $t=3$ . The next time step is  $t=5$ , which is a value in the data base; therefore, no interpolation is required.



**Figure 11. Sample Data: Bandwidth as a Function of Time**

Figure 12 shows the processing that occurs within the prototype. The prototype analyzes incoming values from the ASU outputs to determine the differences from one point to the next and the slopes represented by those differences. The point values and tolerance bands are stored in a data file if the set represents training data. If the set represents operational data, the differences and slopes are also computed and compared to the stored values to determine whether or not an alarm should be raised. Once the data sets have been analyzed, the plotting package is called to display the results for the operator. With the plotting package, the operator can control the number of curves and types of data on the display at one time. His control choices identify which data are collected and displayed on the screen.

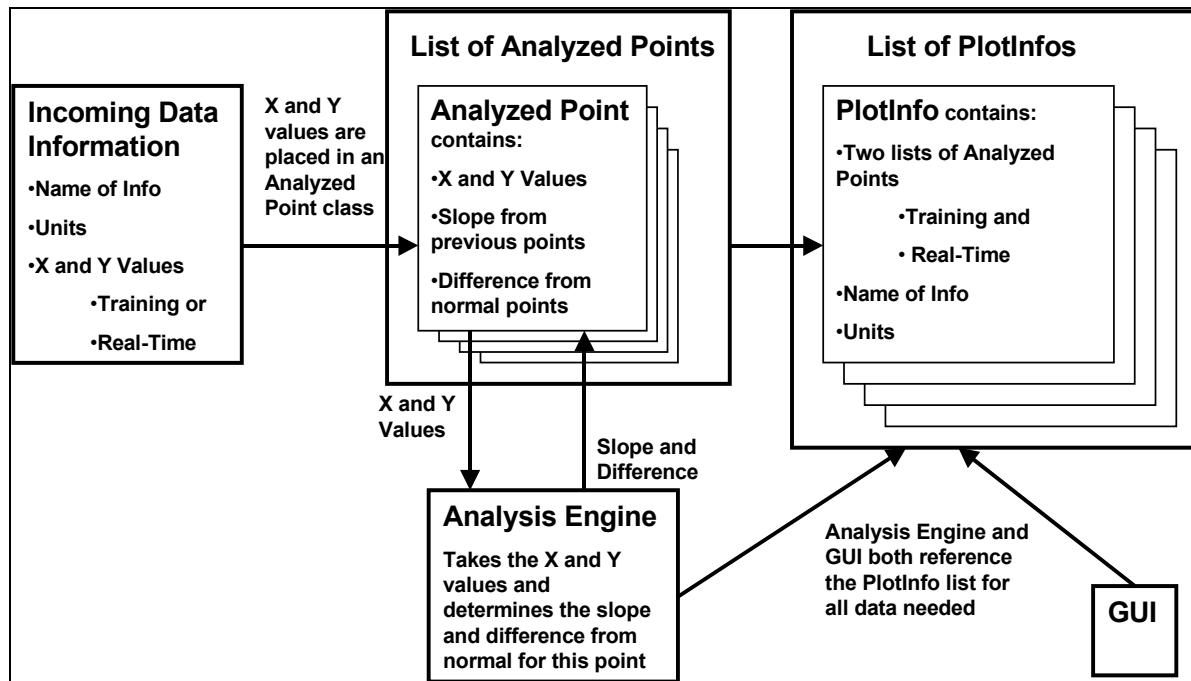
### **4.3 Prototype Results**

The prototype effort resulted in a package that would display data from the behavioral model in various combinations as selected by the user. The effort included development of software tools to better understand the data generated by the simulator at ASU. The elements described below were combined to demonstrate a security operations view of the network and detection of an attack against that network.

#### Develop an architecture for security analysis of an ATM network

The proposal for this past phase of ATM-Sentinel described an architecture for ATM network security. The effort during this phase refined the architecture and developed a simulator to prove the concept of a distributed ATM network security system. Details of the architecture were

described in Paragraph 4.1. The architecture was validated by the prototype and by comparing the architecture to existing network management architecture. Current network architecture



**Figure 12. Prototype processing steps**

models have a set of distributed nodes reporting status via SNMP to a central monitoring station (e.g., HP OpenView)

#### Develop a Parser for the ASU data

To exploit the ASU simulation data, a parser was developed to extract call setup information and link bandwidth data. During the process of extracting the link bandwidth data, the bandwidth that originated or terminated at a node was identified. The data parser required two passes to completely process the node data. The first pass extracted data relative to a single node. A second pass developed the bandwidth information that showed which bandwidth was destined for which node. The data parser was important to put the ASU data into a form that would enable replay of the simulated data that would look like a real network of nodes sending data into the central analysis engine. The data parser was developed as a Symantec VisualCafe native application that will run on an NT4.0 workstation.

#### Develop a simulated ATM switch node that communicates ASU node data to a central processor

A software package capable of simulating ATM cell communication was developed. The simulator reused an IP router simulator previously developed for a different application. The full capabilities of the IP router simulator could be expanded with little effort to simulate each layer of the ATM stack. The switch node simulator allows the development of a network topology that



can be communicated to each of a number of instances of the same software. The software was modified to read the ASU ATM switch node balance data and send it to a central component that collects data from the simulated ATM switches. The ATM switch node software was developed as a Symantec VisualCafe native application that will run on an NT4.0 workstation.

#### Develop a central data collection agent

The central collection agent of the prototype used the basic same software used for the switch nodes (described just above) with the additional processing for collecting and handling its central monitoring functions. When an ATM switch acts as a central node, it will accept messages from the distributed ATM switch nodes and then connect and transfer all data received by it to the analysis engine.

#### Develop a central analysis engine

The key analysis of the ASU data was performed by software within the central analysis engine. The central analysis engine stores training data derived from the ASU normal data. The raw training data are stored in an array to enable the user to view the raw data. The raw training data are also stored in a form to simulate smoothing by a linear interpolation method. Tolerance bands were established for the instantaneous value and first derivative of the training data. The tolerance bands are used when receiving real-time data to alert the operator to an attack. During the network monitoring simulation, real-time data with simulated attacks are transmitted through the system to the analysis engine. The raw real-time data are stored and compared to the tolerance bands. When the analysis engine detects that the real-time data have exceeded the tolerance limits, an alert is issued to the GUI.

The central analysis engine was developed as a Java application using Kawa by Tek-Tools, Inc. The software will run on any platform that runs Java JRE 1.3 The software was demonstrated on an NT4.0 workstation.

#### Develop a graphical user interface to view ATM switch (node) data

A graphical user interface (GUI) was developed to view the training and real-time ATM bandwidth and call-setup data. A special GUI enabled simultaneous display of several graphs on the same abscissa representing time. It provides for multiple ordinates allowing the display of bandwidth, node balance, or call setup information all related to a single node. The GUI enables the user to re-scale the x-axis and all of the y-axes independently. The GUI provides several features that allow selection of various data sets to be displayed with panning and zooming features. The GUI also displays the alerts reported by the analysis engine. In addition to displaying alerts when the real-time data exceeds the thresholds, the curve is shown by a different color when outside the threshold value.

## **5 Summary**

### **5.1 Findings**

The key findings of the ATM-Sentinel Project lie in two major areas. First, the behavioral model and resulting simulation led to a better understanding of the needs of an ATM-Sentinel. Second, the development of an interface that would work with the data produced from the simulation and enable a security operator to monitor effects at several points in the protected network.

Professor Sumit Ghosh at Arizona State University developed the behavioral model that ran on a multi-processor ATM simulation. That simulation modeled the signaling portions of the PNNI protocol. Within the confines of the existing model, the behavioral model demonstrated that there are signatures of the studied attacks. These signatures are typically distributed over the network, even if the attack is focussed on only one part of the network. Thus, the entire network must be monitored if the attacks are going to be discovered while active. Consequently, an ATM-Sentinel must be more than a firewall to provide effective protection of the network. Since it collects data from all over the network, the ATM-S must be designed to minimize the impact on processing time at the nodes and volumes of data transmitted over the network to the central monitor. In fact, for this as well as for security reasons, the ideal design would have the ATM-S operating on its own, separate network. This would protect the security data from attack and avoid the adverse loading or contention on the primary network.

The software prototype was designed to display user selected network statistics to identify when an attack takes place and then to localize it to the appropriate nodes. That display is only one part of a graphical user interface; additionally, the GUI would provide more information about the apparent target and source of the attack. It would also provide details about how to unravel the specifics of the attack to discern its objectives and how best to respond to the attack. As a result, we suggest combining the ATM-S GUI with a tool such as Motorola Intrusion Vision (MIV). MIV assists a security operator to know and respond in near real-time to attacks on a network by providing not only the alerts, but also supplementary data such as the attack details, where else in the network similar attacks have been seen, and the response to the attack.

Collecting and processing of the data for ATM-Sentinel may provide a challenge to the capability of network devices. At each node the QoS statistics must be developed and transmitted to the main monitoring stations. These include the successful message completion and latency time data as well as all of the link statistics that monitor bandwidth usage or availability. Models must be developed for normal data variability and potentially even including time of day, time of month, time of year parameters in that model so that an attack can be recognized as truly abnormal for the anticipated operation. The models must be updated on a regular basis to accommodate the changing nature of the network and its normal traffic load.

### **5.2 Future Work**

#### **5.2.1 Follow-on activities**

Much work remains to design and develop a functioning and practicable ATM Sentinel. The theoretical foundation must be more firmly grounded with consideration of larger networks, better modeling of all aspects of the ATM protocols, better traffic models, and a wider variety of

attacks. Implementation design needs to consider time variability of traffic patterns in setting the thresholds and attack alert conditions. Matching of the network quality of service measurements with more traditional node-based intrusion detection systems that evaluate the re-assembled message actions is another area for development.

Once the foundation is established, attack signatures need to be developed and implemented. One of the significant challenges will be to identify and implement a strategy for making those signatures dynamic in the sense that they respond to changing “normal” conditions on the network, but do not change so rapidly as to mask slow attack activities.

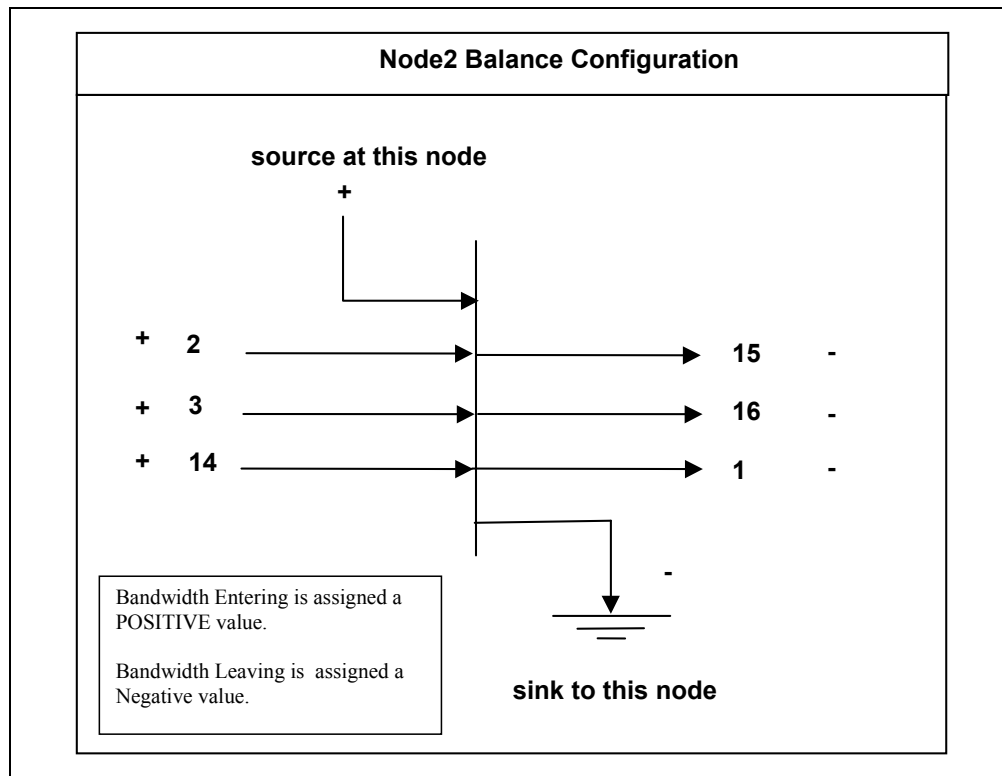
While much of the analysis starts at the nodes, and even some intrusion detection can take place there, the majority of the work will be done at the central monitor station or ATM-Sentinel manager. It is there that the statistics are available for comparison with the expected traffic patterns and the changes can be recognized as anomalous. This implies that the security system may be transmitting a considerable volume of data itself. If this is done on the main network, that has two ramifications. First, the volume may adversely impact the bandwidth available to the operational activity on the network. Second, a change in volume may alert an intruder to the possibility of his discovery, which may be contrary to the desires of the security manager who is trying to learn the intruder’s objectives. After detection, the ATM-Sentinel manager must then implement, automatically or interactively by the security manager, measures to deal with the intruder. These may be as simple as dropping that session, or blocking all sessions from that source for some time period, or as complex as diverting the session to a specially prepared area that mimics the desired target but is specially hardened.

### **5.2.2 *ATM-Sentinel Development***

All of the processing may become a bottleneck at the enclave entrance and slow the data rate to an unacceptable level; however, until we have a better definition of the specific requirements for the ATM Sentinel, we cannot quantify the effect. As the development progresses and data become available from a Behavioral Model to evaluate specific requirements, we will be able to better assess this issue. Another option that we have considered involves distributing some of the ATM Sentinel functions to each individual workstation. This concept, which we have called the ATM Workstation Boundary Controller, provides similar functionality at each node of the enclave. It transmits its information to a security management system that combines the individual reports and implements security policy for the enclave as appropriate for those reports. The communication channel could be over the main network, or, even better, over a separate security channel. While this concept is not currently part of the ATM Sentinel project, it appears to provide a means to reduce the data rate impact of maintaining all of the processing at the enclave boundary.

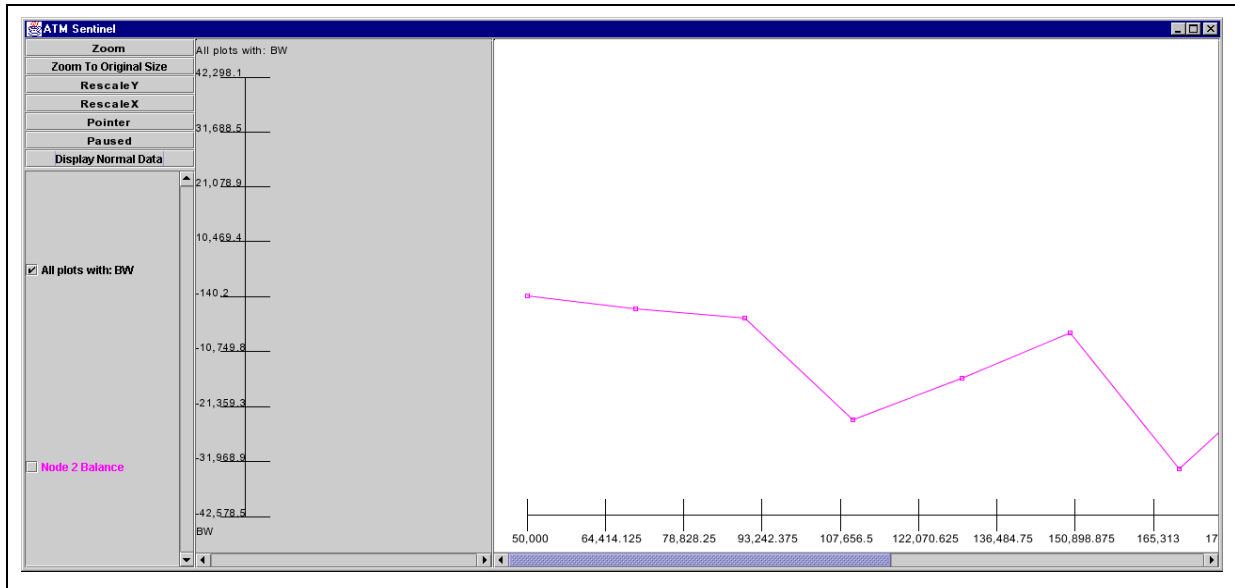
## Appendix A Prototype Display Development

For each node, we evaluated the volume of traffic carried on the links entering and leaving that node and the calls originating and terminating at the node. The analysis results in a diagram illustrated in Figure 13 for Node2. The requested bandwidth related to a call on a link coming into a node is assigned a POSITIVE value. The requested bandwidth on a link related to a call going out of a node is assigned a NEGATIVE value. Calls that originate at the node are assigned POSITIVE values so that they balance the negative outflow on a link. Calls terminating at the node likewise are assigned NEGATIVE values to balance the incoming volume on a link. This node balance is a way to account for all bandwidth on all links connected to the node.



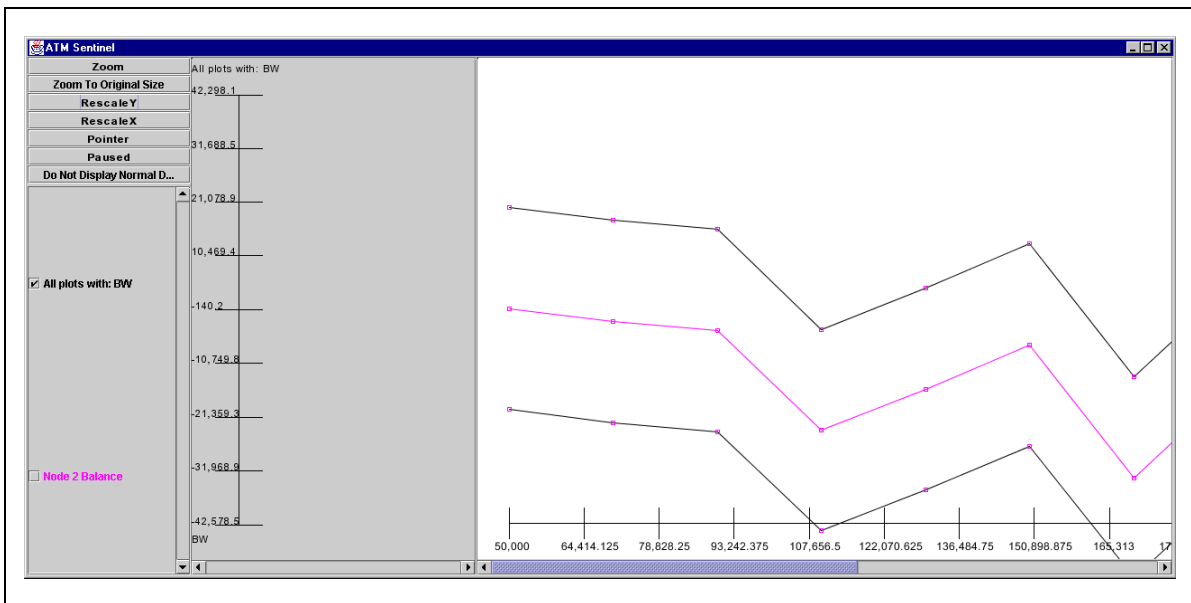
**Figure 13 Node Balance Calculation Includes Data In, Out, Source, and Sink at a Node**

Figure 14 shows the node balance data for Node2 during the training run. Node balance for any node should stabilize around zero. However, due to link delay and the method for accounting for a call completion, the node balance will vary. To account for this variability and the variability due to normal traffic variation, the data should be smoothed before developing tolerance bands for acceptable variability. (Because of the small sample sizes from the Behavioral Model, no smoothing was done for this prototype.) An attack that affects the node traffic or the volume of traffic passing through the node will be seen in a changed value of the node balance.



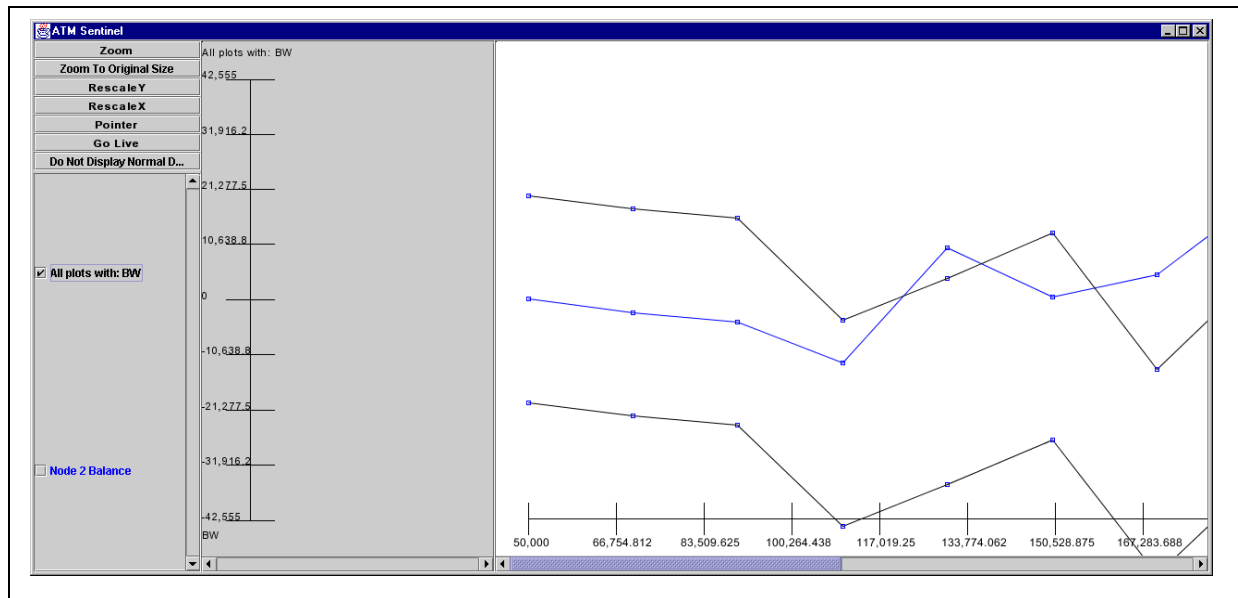
**Figure 14 Display of Node2 Balance: Training (Normal) Data**

Figure 15 shows Node2 data with the tolerance band limits added to the normal data. The lighter curve is the training (normal) data and the upper and lower curves are the upper and lower tolerance values. The GUI provides the capability to display training data, tolerance bands and real-time data on the same plot.



**Figure 15 Training (Normal) Data with tolerance bands**

Figure 16 shows the real-time data plotted with the tolerance bands. Note that the real-time plot goes out of the acceptable region at two different times. The operator would be alerted by email, audio, or some other means when the real-time data goes outside the tolerance bounds.



**Figure 16 Real-Time Data with Threshold Crossing**